Logik für Informatiker
Logic for computer scientists

Till Mossakowski

WiSe 2013/14

Resolution

Recall: Conjunctive Normal Form (CNF)

For each propositional sentence, there is an equivalent sentence of form

$$(\varphi_{1,1} \vee \ldots \vee \varphi_{1,m_1}) \wedge \ldots \wedge (\varphi_{n,1} \vee \ldots \vee \varphi_{n,m_n}) \quad (n \geq 1, m_i \geq 1)$$

where the $\varphi_{i,j}$ are literals, i.e. atomic sentences or negations of atomic sentences.
Note that n may be 1, e.g. $A \vee B$ is in CNF.
Note that the m_i may be 1, e.g. A as well as $A \wedge B$ are in CNF.

A sentence in CNF is called a Horn sentence, if each disjunction of literals contains at most one positive literal.

Examples of Horn sentences

$$\neg Home(claire) \wedge (\neg Home(max) \vee Happy(carl))$$

$$Home(claire) \wedge Home(max) \wedge \neg Home(carl)$$

$$Home(claire) \vee \neg Home(max) \vee \neg Home(carl)$$

$$Home(claire) \wedge Home(max) \wedge$$
$$(\neg Home(max) \vee \neg Home(max))$$

Examples of non-Horn sentences

$$\neg Home(claire) \wedge (Home(max) \vee Happy(carl))$$

$$(Home(claire) \vee Home(max) \vee \neg Happy(claire))$$
$$\wedge Happy(carl)$$

$$Home(claire) \vee (Home(max) \vee \neg Home(carl))$$

Alternative notation for the conjuncts in Horn sentences

$$
\begin{array}{lcl}
\neg A_1 \vee \ldots \vee \neg A_n \vee B & \Leftrightarrow & (A_1 \wedge \ldots \wedge A_n) \rightarrow B \\
\neg A_1 \vee \ldots \vee \neg A_n & \Leftrightarrow & (A_1 \wedge \ldots \wedge A_n) \rightarrow \bot \\
B & \Leftrightarrow & \top \rightarrow B \\
\bot & \Leftrightarrow & \square
\end{array}
$$

Any Horn sentence is equivalent to a conjunction of conditional statements of the above four forms.

Satisfaction algorithm for Horn sentences

1. For any conjunct $\top \rightarrow B$, assign TRUE to B.
2. If for some conjunct $(A_1 \wedge \ldots \wedge A_n) \rightarrow B$, you have assigned TRUE to $A_1, \ldots, A_n$ then assign TRUE to B.
3. Repeat step 2 as often as possible.
4. If there is some conjunct $(A_1 \wedge \ldots \wedge A_n) \rightarrow \bot$ with TRUE assigned to $A_1, \ldots, A_n$, the Horn sentence is not satisfiable. Otherwise, assigning FALSE to the yet unassigned atomic sentences makes all the conditionals (and hence also the Horn sentence) true.

Correctness of the satisfaction algorithm

Theorem The algorithm for the satisfiability of Horn sentences is correct, in that it classifies as tt-satisfiable exactly the tt-satisfiable Horn sentences.

Propositional Prolog

$$
\begin{array}{l}
AncestorOf(a,b) :- MotherOf(a,b). \\
AncestorOf(b,c) :- MotherOf(b,c). \\
AncestorOf(a,b) :- FatherOf(a,b). \\
AncestorOf(b,c) :- FatherOf(b,c). \\
AncestorOf(a,c) :- AncestorOf(a,b), AncestorOf(b,c). \\
MotherOf(a,b). \qquad FatherOf(b,c). \qquad FatherOf(b,d).
\end{array}
$$

To ask whether this database entails B, Prolog adds $\bot \leftarrow B$ and runs the Horn algorithm. If the algorithm fails, Prolog answers "yes", otherwise "no".

Clauses

A clause is a finite set of literals.
Examples:

$$C_1 = \{Small(a), Cube(a), BackOf(b, a)\}$$
$$C_2 = \{Small(a), Cube(b)\}$$
$$C_3 = \emptyset \quad (\text{ also written } \square)$$

Any set $\mathcal{T}$ of sentences in CNF can be replaced by an equivalent set $\mathcal{S}$ of clauses: each conjunct leads to a clause.

Resolution

A clause R is a resolvent of clauses C_1, C_2 if there is an atomic sentence A with $A \in C_1$ and $(\neg A) \in C_2$, such that

$$R = (C_1 \setminus \{A\}) \cup (C_2 \setminus \{\neg A\}).$$

Resolution algorithm: Given a set $\mathcal{S}$ of clauses, systematically add resolvents. If you add $\Box$ at some point, then $\mathcal{S}$ is not satisfiable. Otherwise (i.e. if no further resolution steps are possible and $\Box$ has not been added), it is satisfiable.

Example

We start with the CNF sentence:

$$\neg A \wedge (B \vee C \vee B) \wedge (\neg C \vee \neg D) \wedge (A \vee D) \wedge (\neg B \vee \neg D)$$

In Clause form:

$$\{\neg A\},\ \{B, C\},\ \{\neg C, \neg D\},\ \{A, D\},\ \{\neg B, \neg D\}$$

Apply resolution:

$$\dfrac{\dfrac{\{A,D\} \quad \{\neg A\}}{\{D\}} \qquad \dfrac{\dfrac{\{B,C\} \quad \{\neg C,\neg D\}}{\{B,\neg D\}} \quad \{\neg B,\neg D\}}{\{\neg D\}}}{\Box}$$

Soundness and completeness

Theorem Resolution is sound and complete. That is, given a set $\mathcal{S}$ of clauses, it is possible to arrive at $\square$ by successive resolutions if and only if $\mathcal{S}$ is not satisfiable.

This gives us an alternative sound and complete proof calculus by putting

$$\mathcal{T} \vdash S$$

iff with resolution, we can obtain $\square$ from the clausal form of $\mathcal{T} \cup \{\neg S\}$.